

Arquitetura de Computadores - 2021.1 - Lista de Exercícios
Segunda Avaliação - Assembly x86 Modo Real

1. O arquivo valores.txt (<http://www.lia.ufc.br/~yuri/20211/arquitetura/valores.txt>) possui uma string com uma lista de números separados por vírgulas. Faça um programa que carregue os valores desse arquivo como números inteiros e, em modo gráfico, para cada coluna, pinte de branco o pixel com a linha correspondente ao valor no arquivo lido em sequência. Por exemplo, os três primeiros valores são 100, 104 e 109, então você deve pintar de branco os pixels com coordenadas (0, 100), (1, 104) e (2, 109) e assim por diante. O arquivo possui 320 valores (ou seja, um valor por coluna no modo gráfico VGA 320x200) e no final possui espaços em brancos para completar um tamanho em bytes que seja múltiplo de 512 (ou seja, já está no ponto de concatenar com o seu programa).

2. Escreva um programa para fazer o Olá Mundo. Entretanto, tal programa deve estar no segundo setor do disco, sendo necessário um programa principal, no primeiro setor, responsável apenas por carregá-lo e executá-lo (fazer o jmp). Ou seja, o programa principal deve estar no primeiro setor do disco e, portanto, ter 512 bytes e terminar com os dois bytes de indicação de boot. O segundo programa deve estar no segundo setor e ocupá-lo completamente também para que, juntos, tenham 1024 bytes e a virtualbox possa reconhecer o arquivo binário como imagem de disco. Você pode fazer dois binários separados e depois concatená-los ou fazer um asm só cuja montagem gere o binário completo.

3. Escreva um programa que, dadas duas strings de entrada, sendo a primeira com exatamente 10 caracteres e a segunda com qualquer quantidade de caracteres menor que 10, imprime se a segunda string é ou não substring da primeira.

4. Escreva um programa que mude a paleta de cores para ter as cores em sequência da seguinte forma:

Índice = R,G,B

0 = 0,0,0

1 = 1,0,0

2 = 2,0,0

...

63 = 63,0,0

64 = 0,1,0

65 = 0,2,0

...

127 = 0,63,0

128 = 0,0,1

129 = 0,0,2

...

191 = 0,0,63

192 = 1,1,0

193 = 2,2,0

... 255 = 63,63,0

Feito isso, pinte a tela com cada linha tendo a cor igual a seu número (linha 0 = cor 0, linha 1 = cor 1...).

5. Escreva um programa que apresenta a seguinte saída, perguntando ao usuário o número máximo (no exemplo, 9). Este número deve ser sempre ímpar e o seu programa deve repetir o pedido de entrada se o usuário entrar um valor par.

1 2 3 4 5 6 7 8 9

2 3 4 5 6 7 8

3 4 5 6 7

4 5 6

5

6. Escreva um programa para ler o número de matrícula de uma pessoa. Caso a matrícula seja existente, exiba o nome da pessoa. Caso contrário, mostre a mensagem "Pessoa não matriculada". Você deve usar o arquivo de matrículas disponibilizado, que possui matrículas e nomes separados por ponto e vírgula e um estudante por linha. Ele deve ser carregado pelo seu programa (ou seja, concatene o arquivo ao binário do seu programa usando o comando cat no linux ou type no windows) e ao final das matrículas ele possui espaços em branco para completar os 512 bytes necessários para montar a imagem de disco.
Download: <http://www.lia.ufc.br/~yuri/20211/arquitetura/matriculas.txt> .

7. Dizemos que um inteiro positivo n é perfeito se for igual à soma de seus divisores positivos diferentes de n . Exemplo: 6 é perfeito, pois $1+2+3 = 6$. Escreva um programa que, dado um número, informa se ele é ou não perfeito.